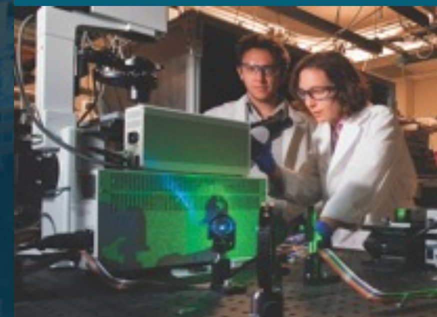
Stanford ME469: Splitting and Stabilization Errors



PRESENTED BY

Stefan P. Domino

Computational Thermal and Fluid Mechanics

Sandia National Laboratories SAND2018-4536 PE

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Splitting and Stabilization Errors: Outline

- Block Matrix and Operator Form
- Approximate Factorization
- Splitting Errors
- Stabilization Errors
- Detailed Code Verification
- Conclusions

Standard Incompressible Equation Set

- Consider the uniform density, low-Mach equation system:

$$\frac{\partial u_j}{\partial x_j} = 0$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} = -\frac{\partial p_i}{\partial x_i} + S_i$$

Standard Incompressible Equation Set

- Consider the uniform density, low-Mach equation system:

$$\frac{\partial u_j}{\partial x_j} = 0$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} = -\frac{\partial p_i}{\partial x_i} + S_i$$

Time Step Count: 8 Current Time: 0.32
dtN: 0.04 dtNm1: 0.04 gammas: 1.5 -2 0.5
Max Courant: 0.22364 Max Reynolds: 1294.89 (realm_1)
Realm Nonlinear Iteration: 1/1

realm_1::advance_time_step()

NLI	Name	Linear Iter	Linear Res	NLinear Res	Scaled NLR
---	----	-----------	----------	-----------	----------
1/2	Equation System Iteration				
1/1	myLowMach				
	MomentumEQS	3	2.62985e-06	0.00890841	1
	ContinuityEQS	4	1.06209e-06	0.0021837	1
2/2	Equation System Iteration				
1/1	myLowMach				
	MomentumEQS	3	2.28651e-08	4.62336e-05	0.00518989
	ContinuityEQS	5	5.2886e-09	5.42362e-05	0.0248369

In our simulation studies, we have been solving this system sequentially and within a nonlinear loop:

For (non-linear iteration) {

1. Momentum
2. Continuity

If (converged) break;

}

Today, we learn more about the consequence of this strategy

Introduction to Block Matrix Form

- Consider the divergence of a vector, **D**F (a scalar)

$$\frac{\partial F_j}{\partial x_j} = 0 \quad \begin{cases} \displaystyle\int \frac{\partial F_j}{\partial x_j} dV = 0 \xrightarrow{\text{(Gauss-Divergence)}} \int F_j n_j \, dS = 0 \\ \displaystyle\int w \frac{\partial F_j}{\partial x_j} d\Omega = 0 \xrightarrow{\text{(Integration-by-parts)}} -\int F_j \frac{\partial w}{\partial x_j} d\Omega + \int w F_j n_j d\,\Gamma = 0 \end{cases}$$

(piecewise-const w)

- Gradient of a scalar, **G**p (a vector)

$$\frac{\partial p}{\partial x_i} = 0 \quad \begin{cases} \displaystyle\int \frac{\partial p}{\partial x_i} dV = 0 \xrightarrow{\text{(Green-Gauss)}} \int p n_i \, dS = 0 \\ \displaystyle\int w \frac{\partial p}{\partial x_i} d\Omega = 0 \xrightarrow{\text{(Integration-by-parts)}} -\int p \frac{\partial w}{\partial x_i} d\Omega + \int w p n_i d\,\Gamma = 0 \end{cases}$$

(piecewise-const w)

- Laplace operator, $\mathbf{L}_\lambda$T

$$\frac{\partial q_j}{\partial x_j} = 0 \qquad q_j = -\lambda \frac{\partial T}{\partial x_j} \quad \begin{cases} \displaystyle\int \frac{\partial q_j}{\partial x_j} dV = 0 \xrightarrow{\text{(Gauss-Divergence)}} -\int \lambda \frac{\partial T}{\partial x_j} n_j \, dS = 0 \\ \displaystyle\int w \frac{\partial q_j}{\partial x_j} d\Omega = 0 \xrightarrow{\text{(Integration-by-parts)}} -\int q_j \frac{\partial w}{\partial x_j} d\Omega + \int w q_j n_j d\,\Gamma = 0 \end{cases}$$

(piecewise-const w)

Introduction to Block Matrix Form for Fluids

- Consider the monolithic, uniform density, low-Mach equation system:

$$\frac{\partial u_j}{\partial x_j} = 0$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} = -\frac{\partial p}{\partial x_i} + S_i$$

that can be written in block form as:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix}$$

- We seek to factorize this system via:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{D} & \boldsymbol{B}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{B}_2\boldsymbol{G} \\ 0 & \boldsymbol{I} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{A} & \boldsymbol{A}\boldsymbol{B}_2\boldsymbol{G} \\ \boldsymbol{D} & (\boldsymbol{B}_1 + \boldsymbol{D}\boldsymbol{B}_2\boldsymbol{G}) \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix}$$

the exact factorization can be recovered by defining: $\left\{ \begin{aligned} \boldsymbol{B}_2 &= \boldsymbol{A}^{-1} \\ \boldsymbol{B}_1 &= -\boldsymbol{D}\boldsymbol{B}_2\boldsymbol{G} \end{aligned} \right.$

- $\mathbf{B}_2$ determines the projection time scale, ideally chosen to approximate $\mathbf{A}^{-1}$
- $\mathbf{B}_1$ controls the projection error, ideally chosen to cancel $\mathbf{BD_2G}$

Introduction to Block Matrix Form

- The approximate factorization

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{D} & \boldsymbol{B}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{B}_2\boldsymbol{G} \\ 0 & \boldsymbol{I} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{A} & \boldsymbol{A}\boldsymbol{B}_2\boldsymbol{G} \\ \boldsymbol{D} & (\boldsymbol{B}_1 + \boldsymbol{D}\boldsymbol{B}_2\boldsymbol{G}) \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix}$$

- can be written now as two segregated steps:

Momentum and Continuity: $\begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{D} & \boldsymbol{B}_1 \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$ $\left\{ \begin{aligned} \boldsymbol{A}\hat{u} &= \boldsymbol{f} \\ \boldsymbol{D}\hat{u} + \boldsymbol{B}_1\hat{p} &= 0 \end{aligned} \right.$

Nodal Projection: $\begin{bmatrix} \boldsymbol{I} & \boldsymbol{B}_2\boldsymbol{G} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix}$ $\left\{ \begin{aligned} u^{n+1} &= \hat{u} - \boldsymbol{B}_2\boldsymbol{G}p^{n+1} = 0 \\ p^{n+1} &= \hat{p} \end{aligned} \right.$

- This approach seems to be straight forward, however, what errors have we introduced by this procedure of splitting the monolithic (fully coupled) system?

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} + \begin{bmatrix} (\boldsymbol{I} - \boldsymbol{A}\boldsymbol{B}_2)\boldsymbol{G}p^{n+1} \\ -(\boldsymbol{B}_1 + \boldsymbol{D}\boldsymbol{B}_2\boldsymbol{G})p^{n+1} \end{bmatrix}$$

Exact iff: $\boldsymbol{B}_2 = \boldsymbol{A}^{-1}$, $\boldsymbol{B}_1 = -\boldsymbol{D}\boldsymbol{B}_2\boldsymbol{G}$

- In most cases, $\mathbf{B}_2$ is approximately $\mathbf{A}^{-1}$ and a first-order temporal splitting error is noted

Recall Mesh Structure… ”co-located”, or “equal-order”

- We have the choice of where the DOFs are solved… Our choice?

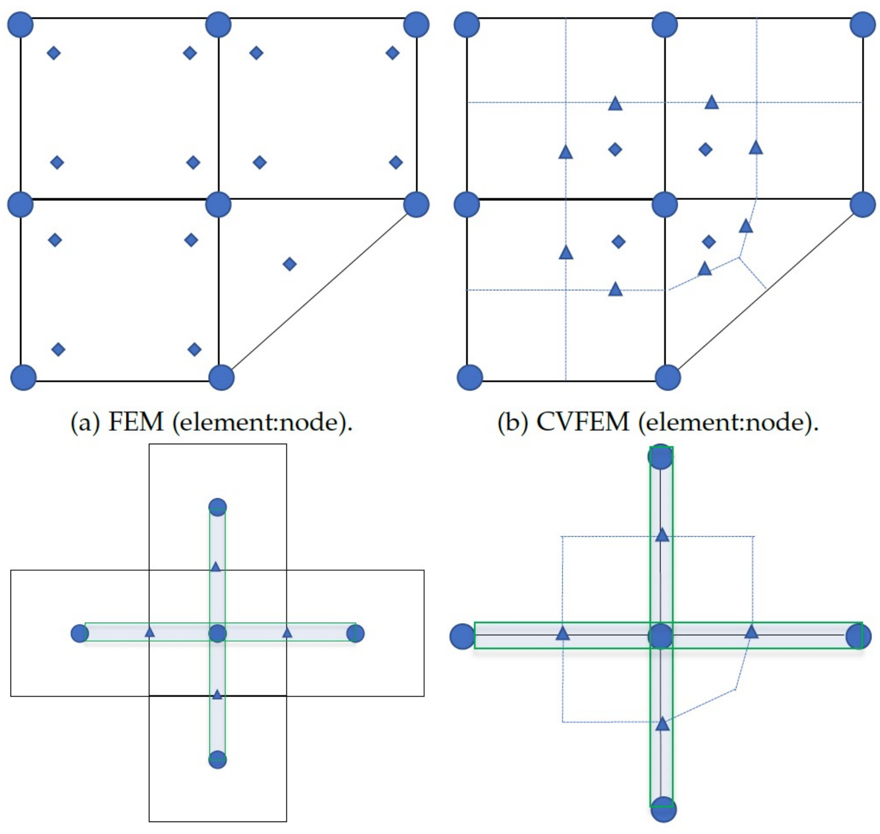
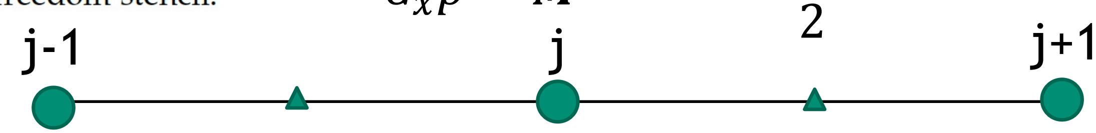


Figure 6: Patches of elements that define center degree-of-freedom stencil.

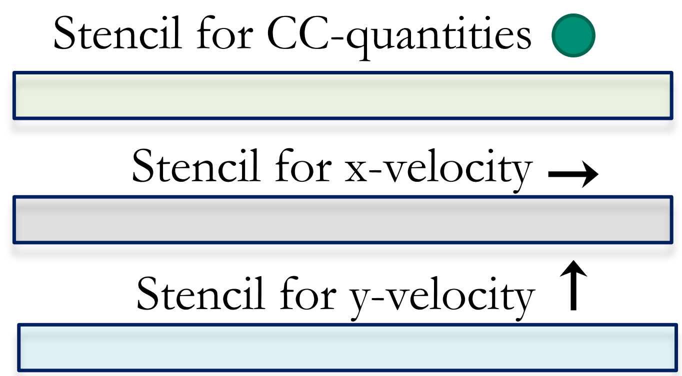
- Here, all variables are solved at the same location (nodes/vertices or cell-centers)
- Underlying basis for each DOF is the same, i.e., “equal-order” interpolation
- We already have noted the gradient operator in 1-D that might pose some problems…

$$\boldsymbol{G}_x p = \frac{(p_{j+1} - p_{j-1})A_x}{2}$$

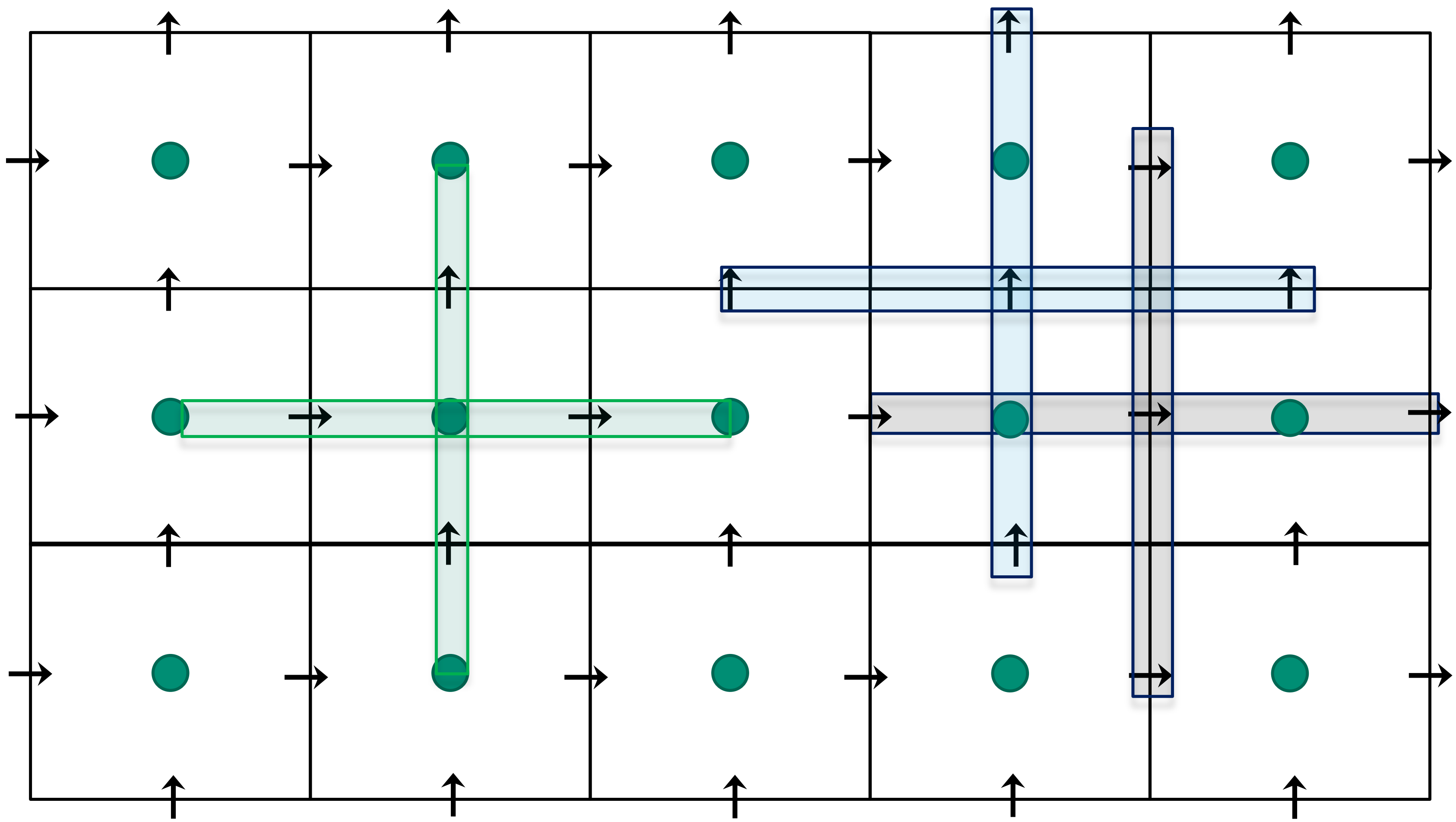
$$G_x p = \boldsymbol{M}^{-1}\frac{(p_{j+1} - p_{j-1})A_x}{2}$$



Staggered…..



- Velocity degree-of-freedom is staggered relative to pressure and other primitives, e.g., enthalpy, mixture fraction, etc.

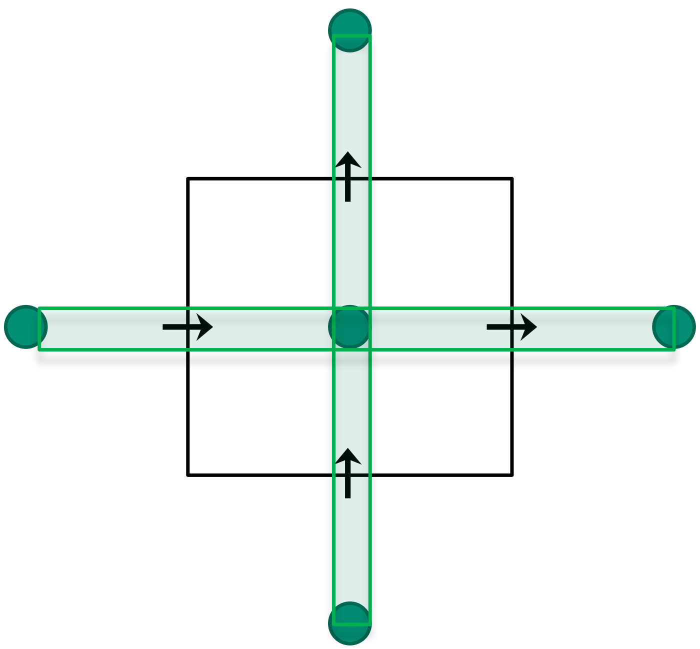


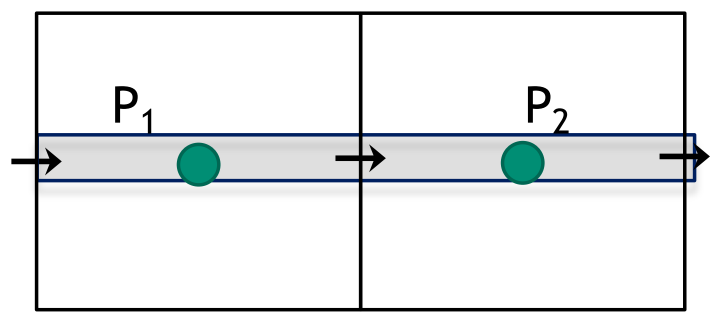
Staggered…..

Stencil for CC-quantities

Stencil for x-velocity $\rightarrow$

- Operators are now very compact and local…





$$G_x p = \boldsymbol{M}^{-1}(p_2 - p_1)A_x$$

Incremental Pressure-Projection without Pressure Stabilization

- Let the inverse of **A**, $\mathbf{A}^{-1}$ be approximated by $\mathbf{B}_2$ as a scalar, $\boldsymbol{\tau}$
- Let $\mathbf{B}_1$ be equal to the scaled Laplace operator, - $\boldsymbol{\tau}\mathbf{L}$

Momentum and Continuity: $\begin{bmatrix} \boldsymbol{A} & \mathbf{0} \\ \boldsymbol{D} & -\tau\boldsymbol{L} \end{bmatrix} \begin{bmatrix} \hat{u} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{f}} \\ -\tau\boldsymbol{L}p^n \end{bmatrix}$ $\left\{ \begin{array}{l} \boldsymbol{A}\hat{u} = \hat{\boldsymbol{f}} = \boldsymbol{f} - \boldsymbol{G}p^n \\ \boldsymbol{D}\hat{u} - \tau\boldsymbol{L}(p^{n+1} - p^n) = 0 \end{array} \right.$

Nodal Projection: $\begin{bmatrix} \boldsymbol{I} & \tau\boldsymbol{G} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} + \begin{bmatrix} \tau\boldsymbol{G}p^n \\ 0 \end{bmatrix}$ $\left\{ \begin{array}{c} u^{n+1} = \hat{u} - \tau G(p^{n+1} - p^n) \\ p^{n+1} = \hat{p} \end{array} \right.$

- The new splitting and stabilization error is given by:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} + \begin{bmatrix} (\boldsymbol{I} - \boldsymbol{A}\tau)\boldsymbol{G}(p^{n+1} - p^n) \\ \tau(\boldsymbol{L} - \boldsymbol{D}\boldsymbol{G})(p^{n+1} - p^n) \end{bmatrix}$$

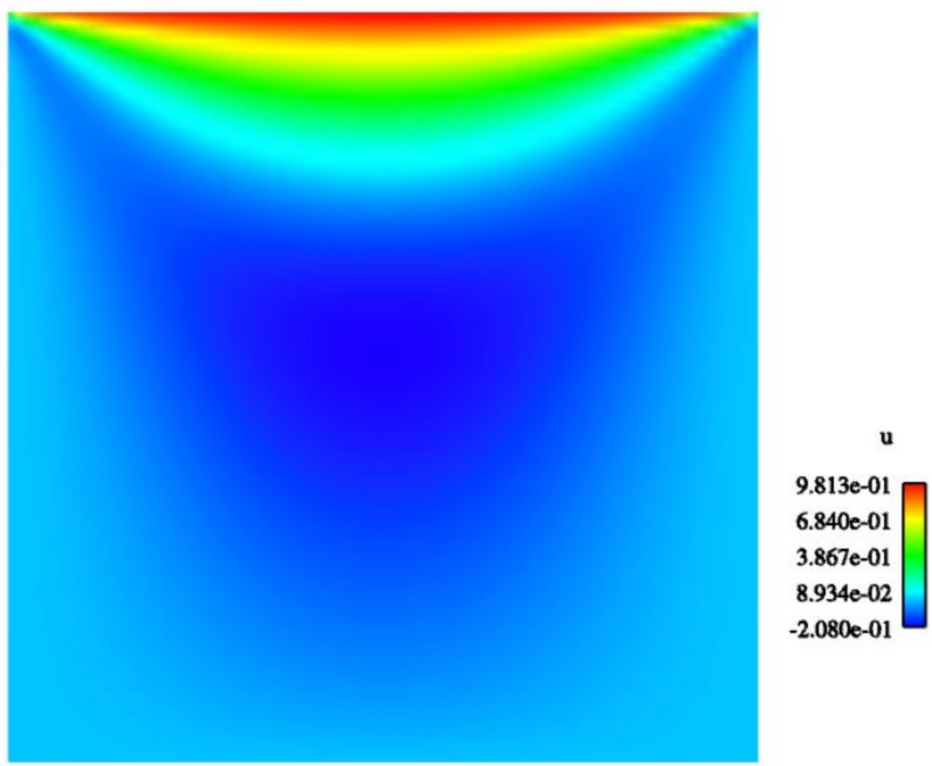
Examples:
- Dwyer (1990)
- Almgren (2000)

- The above can be shown to demonstrate second-order temporal error (coming)
- A scheme can be designed such that **L** = **DG** (staggered)
- A scheme in which **L** != **DG** (collocated or equal-order) can show that **L-DG** ~ 4^{th}-order pressure stabilization, however, in this case acting on p^{n+1}-p^n

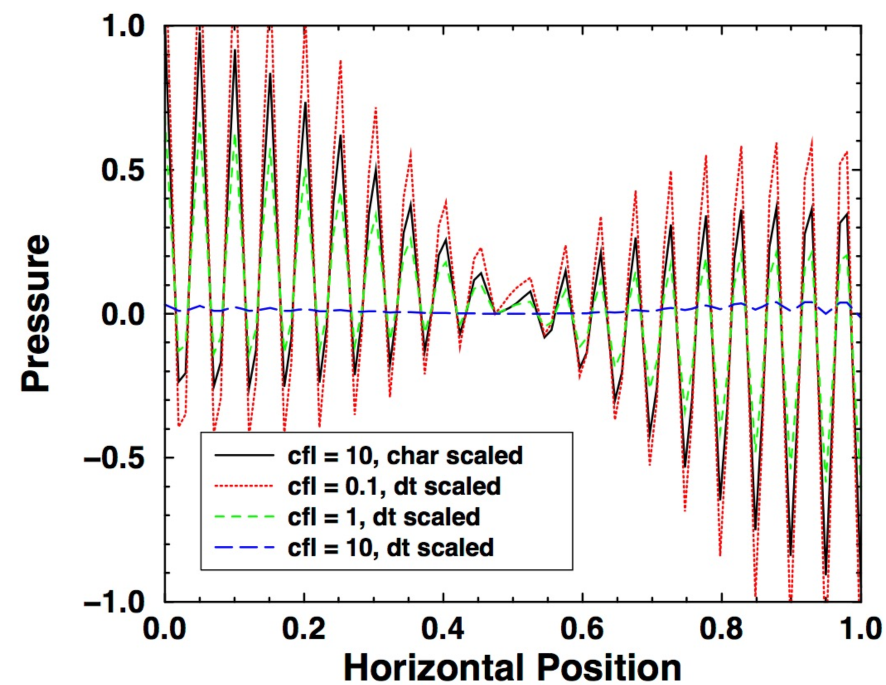
The Role of Pressure Stabilization (L!=DG) in an Equal-Order Approach

- Consider the classic lid-driven cavity flow with top wall velocity of Uo

Uo



lid-driven cavity velocity (u-component)



Unsmoothed pressure field at various Courant numbers

Equal-Order: same basis and interpolation operators for continuity and momentum
Also known as: “collocated”

Incremental Approximate Pressure-Projection with Pressure Stabilization Errors

- Let the inverse of **A**, $\mathbf{A}^{-1}$ be approximated by $\mathbf{B}_2$ as a scalar, **τ (which is ~ time scale)**
- Let $\mathbf{B}_1$ be equal to the scaled Laplace operator, - **τL**

Momentum and Continuity: $$\begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{D} & -\tau\boldsymbol{L} \end{bmatrix} \begin{bmatrix} \hat{u} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{f}} \\ -\boldsymbol{D}\tau\boldsymbol{G}p^{n} \end{bmatrix} \quad \begin{cases} \boldsymbol{A}\hat{u} = \boldsymbol{f} - \boldsymbol{G}p^{n} \\ \boldsymbol{D}\hat{u} = \tau(\boldsymbol{L}p^{n+1} - \boldsymbol{D}\boldsymbol{G}p^{n}) \end{cases}$$

Nodal Projection: $$\begin{bmatrix} \boldsymbol{I} & \tau\boldsymbol{G} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} + \begin{bmatrix} \tau\boldsymbol{G}p^{n} \\ 0 \end{bmatrix} \quad \begin{cases} u^{n+1} = \hat{u} - \tau G(p^{n+1} - p^{n}) \\ p^{n+1} = \hat{p} \end{cases}$$

- The new splitting and stabilization error is given by:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} + \begin{bmatrix} (\boldsymbol{I} - \boldsymbol{A}\tau)\boldsymbol{G}(p^{n+1} - p^{n}) \\ \tau(\boldsymbol{L} - \boldsymbol{D}\boldsymbol{G})p^{n+1} \end{bmatrix}$$

Examples:

- Rhie-Chow (1983)
- Peric (1985)
- Domino (2006)

- The above can be shown to hold a second-order temporal error (coming)
- Here, due to equal-order interpolation, i.e., collocation of primitives, **L** != **DG**
- Therefore, **L-DG** ~ 4th-order pressure stabilization (pressure oscillations damped)
- However, pressure-stabilization error remains

Non-Incremental Pressure-Projection: "Pressure-Free" with or without Pressure Stabilization

- Let the inverse of **A**, $\mathbf{A}^{-1}$ be approximated by $\mathbf{B}_2$ as a scalar, $\boldsymbol{\tau}$
- Let $\mathbf{B}_1$ be equal to the scaled Laplace operator, - $\boldsymbol{\tau}\mathbf{L}$

Momentum and Continuity: $$\begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{D} & -\boldsymbol{\tau L} \end{bmatrix} \begin{bmatrix} \hat{u} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} \quad \left\{ \begin{array}{c} \boldsymbol{A}\hat{u} = f \\ \boldsymbol{D}\hat{u} - \tau \boldsymbol{L}(p^{n+1}) = 0 \end{array} \right.$$

Nodal Projection: $$\begin{bmatrix} \boldsymbol{I} & \tau\boldsymbol{G} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} + \begin{bmatrix} \tau\boldsymbol{G}p^{n} \\ 0 \end{bmatrix} \quad \left\{ \begin{array}{c} u^{n+1} = \hat{u} - \tau G(p^{n+1}) \\ p^{n+1} = \hat{p} \end{array} \right.$$

- The new splitting and stabilization error is given by:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} + \begin{bmatrix} (I - \boldsymbol{A}\tau)\boldsymbol{G}(p^{n+1}) \\ \tau(\boldsymbol{L} - \boldsymbol{DG})(p^{n+1}) \end{bmatrix}$$

Examples:

- Kim and Moin (1985)

- A fully-implicit scheme can be shown to demonstrate first-order temporal error (coming)
- A scheme can be designed such that **L** = **DG** (staggered) to remove stabilization error
- A scheme in which **L** != **DG** (collocated or equal-order) can show that **L-DG** ~ 4^{th}-order pressure stabilization (however, in this case acting on p^{n+1}-p^{n}

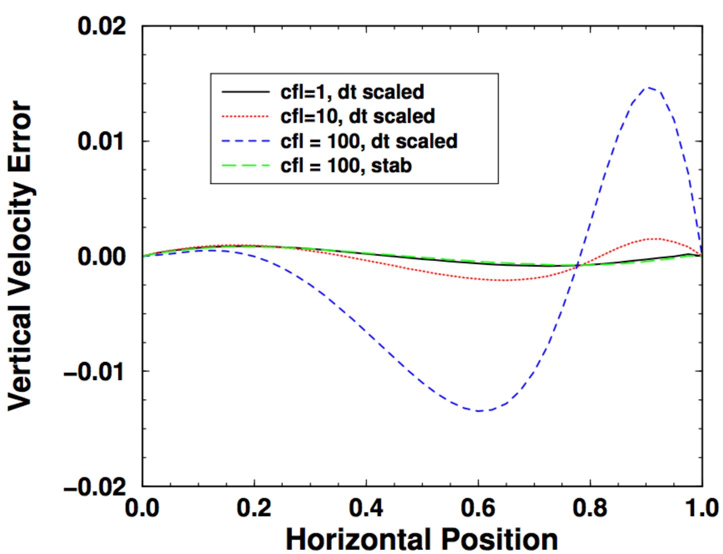
Sensitivity to chosen τ

- Recall that the equal-order pressure stabilization error is given by,

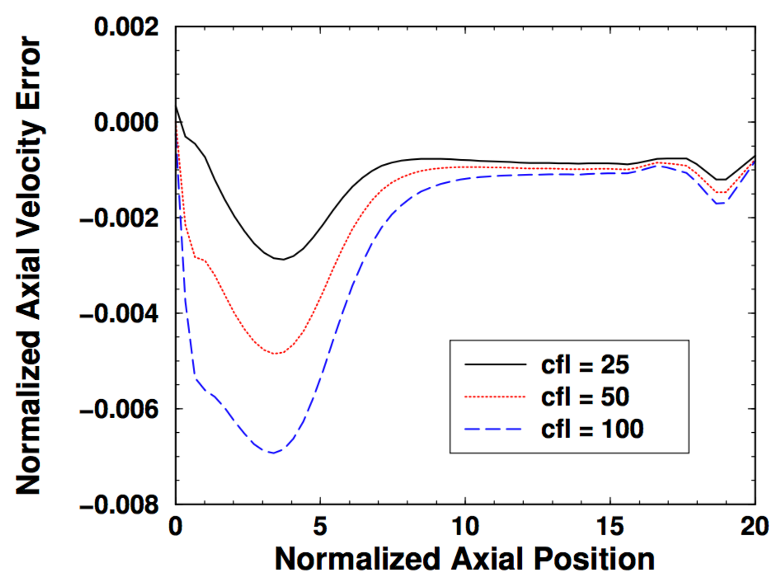
$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} + \begin{bmatrix} (\boldsymbol{I} - \boldsymbol{A}\tau)\boldsymbol{G}(p^{n+1} - p^{n}) \\ \tau(\boldsymbol{L} - \boldsymbol{D}\boldsymbol{G})p^{n+1} \end{bmatrix}$$

Stabilizing effect

- Practical examples of error as a function of:
  - $\tau \propto \Delta t$ (the simulation timestep)
  - $\tau^{C} \propto (u/\Delta x)^{-1}$ (characteristic advection time scale)
  - $\tau^{C}(\boldsymbol{L} - \boldsymbol{D}\boldsymbol{G})p^{n+1} + \Delta t \boldsymbol{L}(p^{n+1} - p^{n})$ (Soto and Lohner, “stabilized”)



Driven Cavity



Open Jet

Monolithic Staggered or Equal-order Interpolation (Collocated)

- Note that we need not split the system for a staggered scheme:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & -\tau\boldsymbol{L} \end{bmatrix}\begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ -\tau\boldsymbol{L}p^n \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix}\begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tau(\boldsymbol{L}-\boldsymbol{D}\boldsymbol{G})(p^{n+1}-p^n) \end{bmatrix}$$

- or collocated:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & -\tau\boldsymbol{L} \end{bmatrix}\begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ -\tau\boldsymbol{D}\boldsymbol{G}p^n \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix}\begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \tau(\boldsymbol{L}-\boldsymbol{D}\boldsymbol{G})p^{n+1} \end{bmatrix}$$

- Conclusion: Monolithic schemes control splitting error, however, dealing with pressure stabilization is an additional complexity for equal-order methods regardless of the chosen approach to solve the coupled system

The Choice of the B_I can Vary by The Method..

- In Pressure-Stabilized Petrov-Galerkin (PSPG) methods (Hughes et al, 1985):

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & -\tau_{PSPG} G q \cdot \boldsymbol{M} \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix}$$

- Here, **M** is a fine scale momentum residual and q is the test function for the Finite Element Method (continuity equation)
- Note that **M** contains a local pressure gradient which, thereby, provides the pressure stabilization
- The fine-scale momentum residual is evaluated locally at the quadrature point and with mesh refinement reduces at a design-order rate
- With some algebra, one can show that **L-DG** ~ **M**
- For more references, see Majumdar, Numerical Heat Transfer, 1988; Tezduyar and Sathe, J. Comput. Appl. Mech., 2003
- Here, τ can be:
  - Simulation time step, i.e., Δt
  - Local advection/diffusion time scale, $\left(\frac{u}{\Delta x} + \frac{v}{\Delta x^2}\right)^{-1}$ - or a more accurate flow-aligned approach
  - Diagonal of A
  - Full inverse of A (Ozawa)

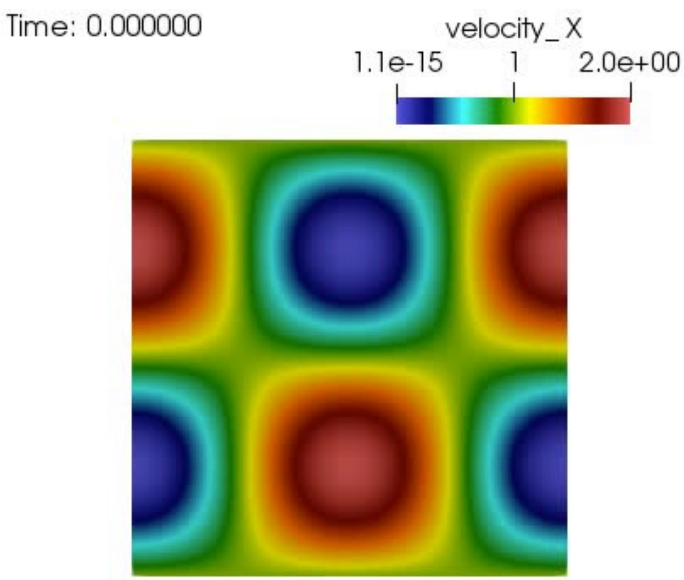
Code Verification To Establish Accuracy

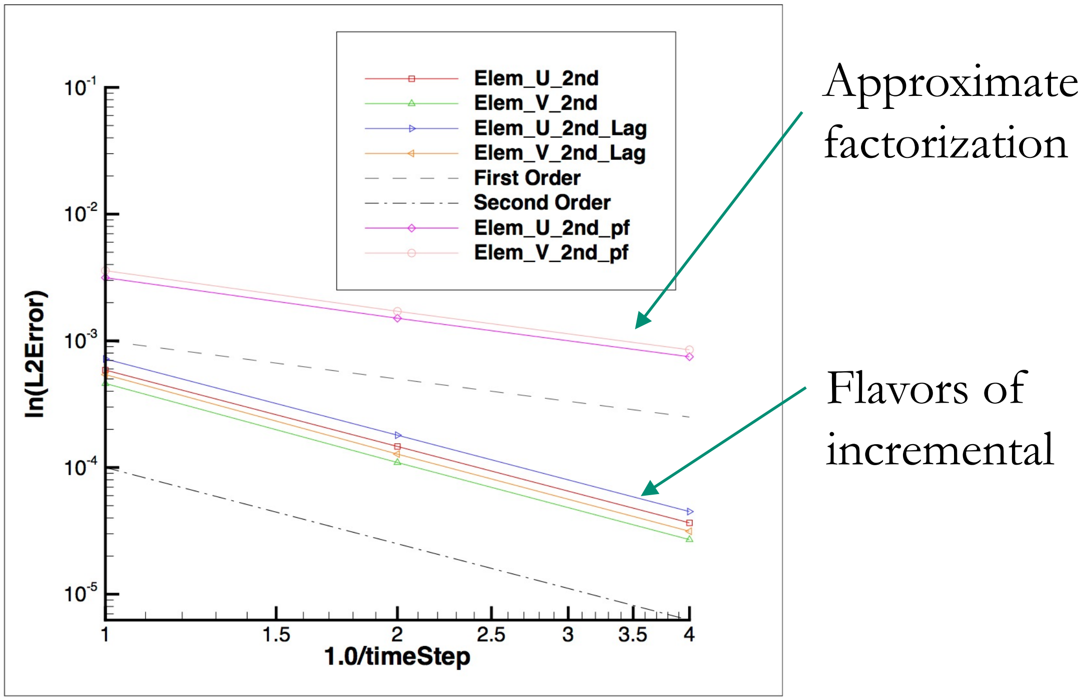
- Consider a two-dimensional transient solution to the incompressible equations of motion:
- This solution is known as the convecting, decaying, Taylor vortex

$$u = u^o - \cos\left(\pi(x - u^o t)\right)\sin\left(\pi(y - v^o t)\right) e^{-2\omega t}$$

$$v = v^o + \sin\left(\pi(x - u^o t)\right)\cos\left(\pi(y - v^o t)\right) e^{-2\omega t}$$

$$p = -\frac{p^o}{4}\left[\,(\cos\left(2\pi(x - u^o t)\right) + \cos(\,\pi(y - v^o t))\right]e^{-4\omega t}$$

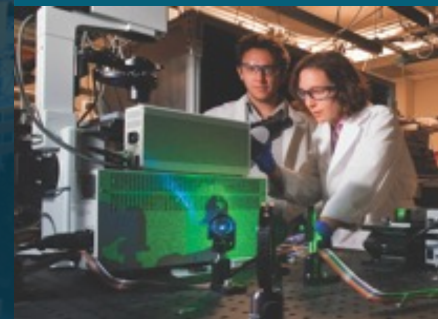




Splitting and Stabilization Errors: Conclusion

- Block matrix and operator form represents a useful construct to analyze coupling and stabilization
- Approximate Factorization is generally $O(\Delta t)$
- With very simple modifications, splitting error is mitigated
- Detailed code verification is a critical tool to both test theoretical understandings in addition to establishing a proper code implementation

Stanford ME469: Splitting and Stabilization Errors Review



PRESENTED BY

Stefan P. Domino

Computational Thermal and Fluid Mechanics

Sandia National Laboratories SAND2018-4536 PE

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Introduction to Block Matrix Form for Fluids

- Consider the monolithic, uniform density, low-Mach equation system:

$$\frac{\partial u_j}{\partial x_j} = 0$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_j u_i}{\partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} = -\frac{\partial p}{\partial x_i} + S_i$$

that can be written in block form as:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix}$$

- We seek to factorize this system via:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{D} & \boldsymbol{B}_1 \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{B_2 G} \\ 0 & \boldsymbol{I} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{A} & \boldsymbol{A B_2 G} \\ \boldsymbol{D} & (\boldsymbol{B_1} + \boldsymbol{D B_2 G}) \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix}$$

the exact factorization can be recovered by defining: $\left\{ \begin{aligned} \boldsymbol{B_2} &= \boldsymbol{A}^{-1} \\ \boldsymbol{B_1} &= -\boldsymbol{D B_2 G} \end{aligned} \right.$

- $\mathbf{B}_2$ determines the projection time scale, ideally chosen to approximate $\mathbf{A}^{-1}$
- $\mathbf{B}_1$ controls the projection error, ideally chosen to cancel $\mathbf{BD_2G}$

Coupling Definitions: Monolithic

Fully Coupled, or Monolithic

- One equation system, one solve

For a given time step:

do while (!converged) {

$$\begin{bmatrix} \frac{\partial}{\partial p}C & \frac{\partial}{\partial \tilde{u}_x}C & \frac{\partial}{\partial \tilde{u}_y}C & \frac{\partial}{\partial \tilde{z}}C \\ \frac{\partial}{\partial p}\tilde{U}_x & \frac{\partial}{\partial \tilde{u}_x}\tilde{U}_x & \frac{\partial}{\partial \tilde{u}_y}\tilde{U}_x & \frac{\partial}{\partial \tilde{z}}\tilde{U}_x \\ \frac{\partial}{\partial p}\tilde{U}_y & \frac{\partial}{\partial \tilde{u}_x}\tilde{U}_y & \frac{\partial}{\partial \tilde{u}_y}\tilde{U}_y & \frac{\partial}{\partial \tilde{z}}\tilde{U}_y \\ \frac{\partial}{\partial p}\tilde{Z} & \frac{\partial}{\partial \tilde{u}_x}\tilde{Z} & \frac{\partial}{\partial \tilde{u}_y}\tilde{Z} & \frac{\partial}{\partial \tilde{z}}\tilde{Z} \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta \tilde{u}_x \\ \Delta \tilde{u}_y \\ \Delta \tilde{z} \end{bmatrix} = - \begin{bmatrix} resC \\ res\tilde{U}_x \\ res\tilde{U}_y \\ res\tilde{Z} \end{bmatrix}$$

}

- Reduced radius of convergence
- Second order convergence

Coupling Definitions: Segregated

Loosely Coupled, or Segregated

- Multiple matrix systems solved within a Picard linearization, i.e., Jacobi or Gauss-Seidel iteration. Coupling is two-way (or multiple way if the physics becomes really complex)

For a given time step:

do while (!converged) {

$$\bar{\rho} = f(\tilde{Z})$$

$$\left[\frac{\partial}{\partial \tilde{u}_x}\tilde{U}_x\right]\left[\Delta\tilde{u}_x\right] = -\left[res\tilde{U}_x\right]$$

$$\left[\frac{\partial}{\partial \tilde{u}_y}\tilde{U}_y\right]\left[\Delta\tilde{u}_y\right] = -\left[res\tilde{U}_y\right]$$

$$\tilde{u}_i^{k+1} = \tilde{u}_i^k + \alpha\Delta\tilde{u}_i$$

$$\left[\frac{\partial}{\partial p}C\right]\left[\Delta p\right] = -\left[resP\right]$$

$$\tilde{u}_i^{n+1} = \tilde{u}_i^{k+1} - \tau\nabla(\Delta p^{n+1/2})$$

$$\left[\frac{\partial}{\partial \tilde{z}}\tilde{Z}\right]\left[\Delta\tilde{z}\right] = -\left[res\tilde{Z}\right]$$

$$\tilde{z}^{k+1} = \tilde{z}^k + \alpha\Delta\tilde{z}$$

}

Every equation is segregated..

- Larger radius of convergence
- Often, stagnates to first-order convergence

Incremental Approximate Pressure-Projection with Pressure Stabilization Errors

- Let the inverse of **A**, $\mathbf{A}^{-1}$ be approximated by $\mathbf{B}_2$ as a scalar, **τ (which is ~ time scale)**
- Let $\mathbf{B}_1$ be equal to the scaled Laplace operator, - **τL**

Momentum and Continuity: $$\begin{bmatrix} \boldsymbol{A} & 0 \\ \boldsymbol{D} & -\tau\boldsymbol{L} \end{bmatrix} \begin{bmatrix} \hat{u} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{f}} \\ -\boldsymbol{D}\tau\boldsymbol{G}p^n \end{bmatrix} \left\{ \begin{aligned} \boldsymbol{A}\hat{u} &= \boldsymbol{f} - \boldsymbol{G}p^n \\ \boldsymbol{D}\hat{u} &= \tau(\boldsymbol{L}p^{n+1} - \boldsymbol{D}\boldsymbol{G}p^n) \end{aligned} \right.$$

Nodal Projection: $$\begin{bmatrix} \boldsymbol{I} & \tau\boldsymbol{G} \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} + \begin{bmatrix} \tau\boldsymbol{G}p^n \\ 0 \end{bmatrix} \left\{ \begin{aligned} u^{n+1} &= \hat{u} - \tau G(p^{n+1} - p^n) \\ p^{n+1} &= \hat{p} \end{aligned} \right.$$

- The new splitting and stabilization error is given by:

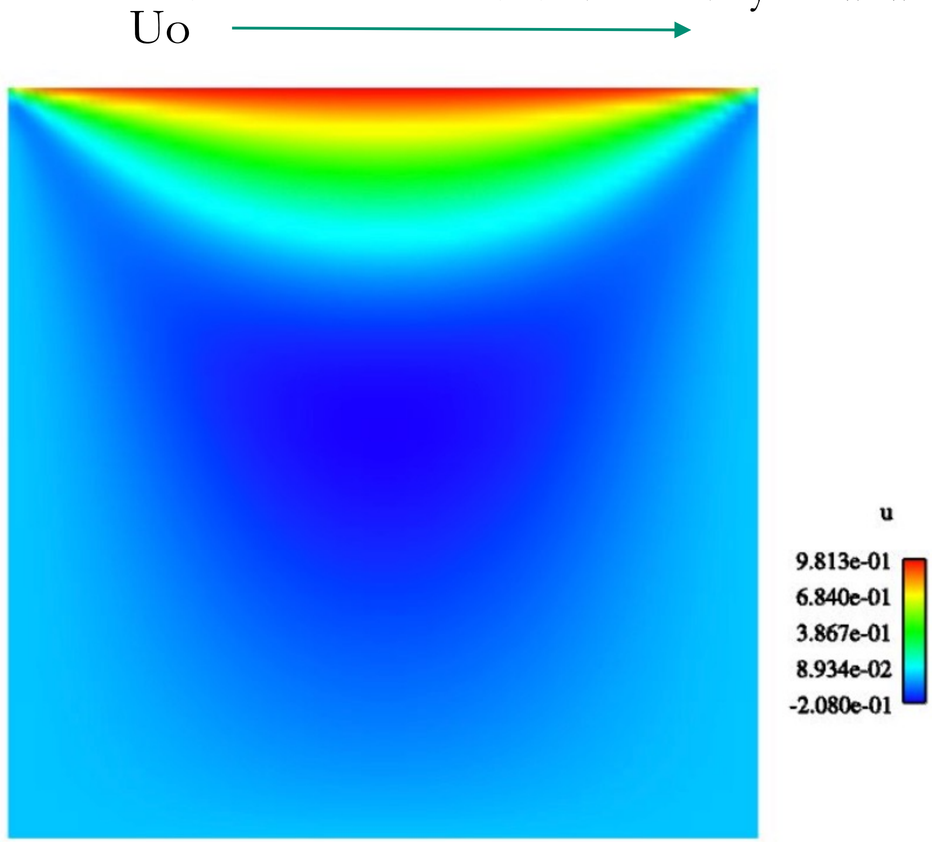
$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{G} \\ \boldsymbol{D} & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} + \begin{bmatrix} (\boldsymbol{I} - \boldsymbol{A}\tau)\boldsymbol{G}(p^{n+1} - p^n) \\ \tau(\boldsymbol{L} - \boldsymbol{D}\boldsymbol{G})p^{n+1} \end{bmatrix}$$

Examples:
- Rhie-Chow (1983)
- Peric (1985)
- Domino (2006)

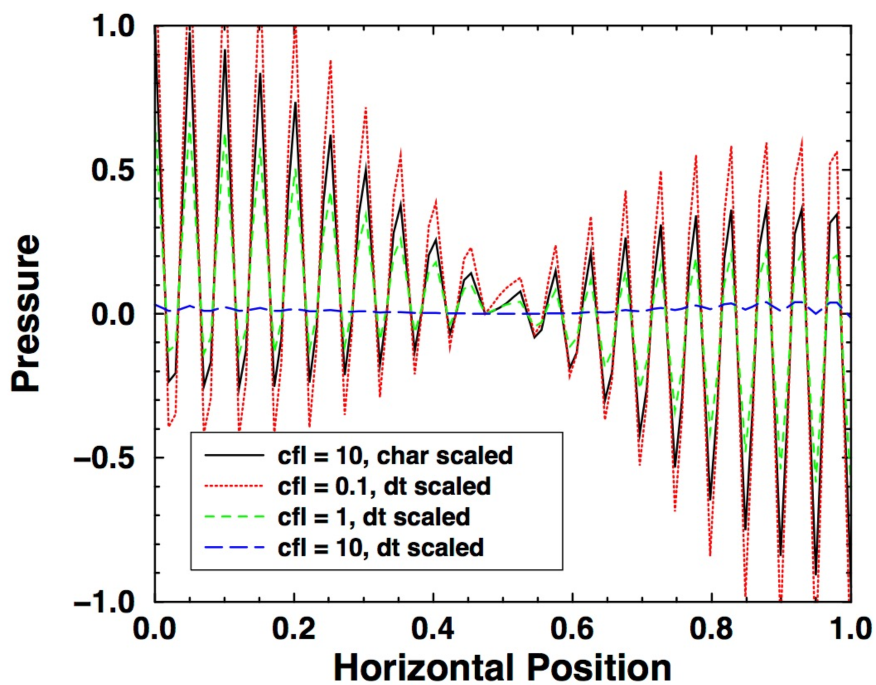
- The above can be shown to hold a second-order temporal error (coming)
- Here, due to equal-order interpolation, i.e., collocation of primitives, **L** != **DG**
- Therefore, **L-DG** ~ 4th-order pressure stabilization (pressure oscillations damped)
- However, pressure-stabilization error remains

The Role of Pressure Stabilization (L!=DG) in an Equal-Order Approach

- Consider the classic lid-driven cavity flow with top wall velocity of Uo



lid-driven cavity velocity (u-component)



Unsmoothed pressure field at various Courant numbers

Equal-Order: same basis and interpolation operators for continuity and momentum

Also known as: “collocated”

First, The Role of $\dot{m}$

- For an equal-order, low-Mach approximate projection scheme, explicit pressure stabilization was added. How does this manifest itself in the advection operator?

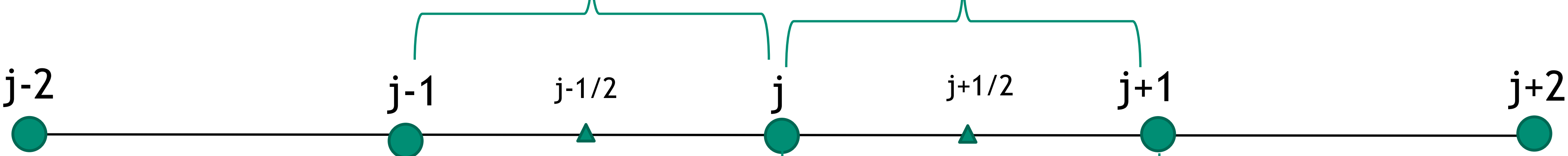
$$\int w \frac{\partial \rho u_j Z}{\partial x_j} dV = \int \rho u_j Z \, n_j \, dS \approx \sum_{ip} \dot{m} Z_{ip}$$

Recall discrete continuity solve

$$\boldsymbol{D}\hat{u} = \tau(\boldsymbol{L}p^{n+1} - \boldsymbol{D}\boldsymbol{G}p^{n})$$

$$\frac{\partial P_{j-1/2}}{\partial x} = \frac{P_j - P_{j-1}}{\Delta x}$$

$$\frac{\partial P_{j+1/2}}{\partial x} = \frac{P_{j+1} - P_j}{\Delta x}$$



$$G_x P_{j-1} = \frac{P_j - P_{j-2}}{2\Delta x}$$

$$G_x P_j = \frac{P_{j+1} - P_{j-1}}{2\Delta x}$$

$$G_x P_{j+1} = \frac{P_{j+2} - P_j}{2\Delta x}$$

- Using the above equations, we can derive the actual continuity equation that we are solving:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} \neq 0 \propto \tau \frac{\partial^4 P}{\partial x_j^4} \Delta x_j{}^3$$

First, The Role of $\dot{m}$

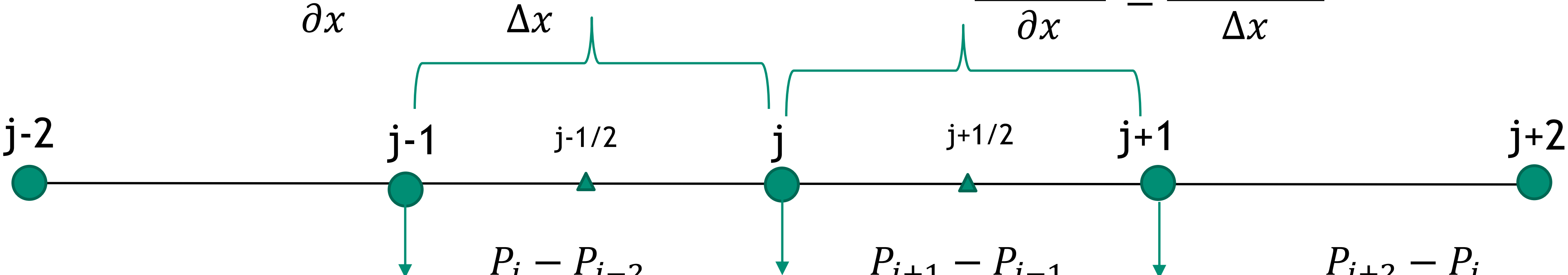
- For an equal-order, low-Mach approximate projection scheme, explicit pressure stabilization was added. How does this manifest itself in the advection operator?

$$\int w \frac{\partial \rho u_j Z}{\partial x_j} dV = \int \rho u_j Z \, n_j \, dS \approx \sum_{ip} \dot{m} Z_{ip}$$

Recall discrete continuity solve

$$\boldsymbol{D}\hat{u} = \tau(\boldsymbol{L}p^{n+1} - \boldsymbol{D}\boldsymbol{G}p^{n})$$

$$\frac{\partial P_{j-1/2}}{\partial x} = \frac{P_j - P_{j-1}}{\Delta x} \qquad \frac{\partial P_{j+1/2}}{\partial x} = \frac{P_{j+1} - P_j}{\Delta x}$$



$$G_x P_{j-1} = \frac{P_j - P_{j-2}}{2\Delta x} \qquad G_x P_j = \frac{P_{j+1} - P_{j-1}}{2\Delta x} \qquad G_x P_{j+1} = \frac{P_{j+2} - P_j}{2\Delta x}$$

- Using the above equations, we can derive the actual continuity equation that we are solving:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} \neq 0 \propto \tau \frac{\partial^4 P}{\partial x_j^4} \Delta x_j{}^3$$

Wait, What is an alternative view of what L-DG forms?

- Let's define an assembled system for velocity component as:

$\mathbf{T}u_i + \mathbf{V}u_i - \mathbf{D}u_i - \mathbf{S} = -\mathbf{G}p_i$

Where **T**, **V**, **D**, **S**, and **G** are the time, advection, diffusion, source and gradient operators (matrix form)

- We can do the same for a local integration point fine-scale momentum equation:

$Tu_i + Vu_i - Du_i - S = -dp/dx_i|ip$

Algebraically, we can reconstruct the fine scale residual to be an interpolation of the nodal-assembled momentum equation that we solved:

- $(dp/dx_i - \mathbf{M}^{-1}\mathbf{G}p_i|ip) = Res\_u_i$

…. And uses this scaled residual as a correction to the interpolated velocity

- The finite volume community calls this "Rhie-Chow", or "Momentum interpolation", while the finite element community terms this "residual-based stabilization"